数理解析研究所講究録
第461巻 1982年 368-379

D A T A B O X

TOGASI Masatomo
Dept. of Phys.
Faculty of Sci.
Hokkaido Univ.

1. Concept of Databox

Database management systems (DBMS) have become to have many application fields. The major fields have been in bussiness world. DBMS is so designed, at least in its beginning, that it can adapt routine work of inquiring information to a huge file and/or work of updating some portions of the file according to change in "real world". The information processed within such a environments can be projected to machine oriented data structures which ussually are formatted records containing several fields, because it is essentially important that a huge file or a set of huge files can be managed systematically, rather than that preciselly is the information mapped into the files from the "real world". Reletionships among the data used to be represented by concatenation of elementary fileds to make up record and by linkage between the records or files.

For scientific information, howwer, DBMS seems to failed to satisfy needs from researchers working with research institutes and universities. Environments under which the researchers are working have some distinctions which are not necessarily included in bussiness offices. Flow of data, point of data generation, method of data processing, and store of data are all tentative ones, and dynamically change according to their stages of research activities. Futhermore, it often occurs that some unexpected, or

some new, information is found, i.e. is discovered, from stack of data through processing for the purpose of total organization of the information. This means that research oriented system designed to match the environments are essentially productive and evolutional in the sense that the file itself produce new information from it. On the other hand, it can be said that research activity is yet personal relative to that of bussiness. Individual researcher has one's own interest and the way of data management would also vary from person to person. Needless to say, current DBMSs can be applied to this field, if one accept limmitations and uneasiness in practical use. But just in its proper fields of application, the DBMS would work efficiently.

From the above consideration, yet another principle of information management system would be desirable to be set for management of research information. It may have complementary role of that of DBMS.

The principle consists of the following points.

(1) personal information system.
    sharing of data is not necessarily considred.

(2) no homogeneity is assumed.
    schema can not be expected to be prepared.

(3) evolutional system.
    information evolves according to user's work.

(4) set oriented system
    information is reduced by generalization

(5) no access structure is added to data structure.
    data is accessed through its own structure.

We call the system based upon the principle as 'databox'.

2. Infomation Structuring.

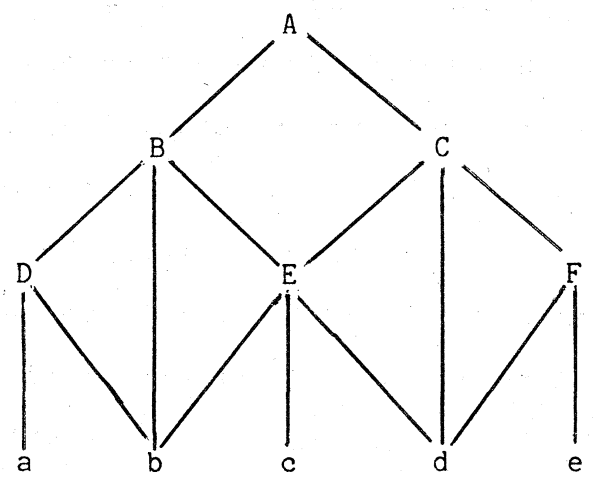
As we do in formation of concepts or in pigeonholing data, clustering of items which are similar each other used bto be a common way. The clustered items as a whole is regarded as a single item, and is attatched by a name by which one can recall its contents. This makes it easy to manage huge number of items in our memory, because we has to be conscious only of these abstuct items in many cases. By repeating the process of formation of cluster i.e. set, a sort of hierarchical organization of information is made in our mind. The same can be said for management of documents, data sheets and data in computer file. Introducing hierarchy in management of various things is so natural that it would become one of basic ideas for manegement of scientific information which is not yet well organized as formatted files. By naturality, we mean that it doesn't impose any specific procedure other than those which are employed in handling documents or notebooks in laboratories: We sort them, attach tiles and dates and so on to them and place them on somewhere easy to find. The organization of information in databox is based on notion of set which is formed hierachically.

On the other hand, many techniques of data organization are employed in DBMS. Among them, inverted file stucture is important in connection with set formation. Usually inverted file is regarded as redundant information in that it doesn't convey any information and it only gives access paths to taget file. However, if we interpret each entry of the inverted file as

definition of a set, it just conveys certain information which, at least, is not explicitly included in the target file. Furthermore, entry keys of the inverted file are not necessarily directly derived, like keywords, from original records. Any human criteria can be used for formation of sets, and keys corresponding each criteria may be used as entry keys of the inverted file.

Thus the inverted file structure is found to be able to have a positive role in representing information. An extention of this stucture is made in databox to let it convey whole information: The entries of inverted file is again regarded as base set for inversion. In other words, nesting of set formation is employed as the extention. The nested set stucture includes the inverted file structure as its most simple form.

Let us define 'primary set' as set of atomic information which is not more reducible, and 'secondary set' as set derived by set operations on the primary set. A set may be an element for sets of upper levels. So we use a word 'node' to specify it neglecting difference of set and element. (Fig. 1)



A = (B, C)

B = (D, b, E)

C = (E, d, F)

D = (a, b)

E = (b, c, d)

F = (d, e)

Fig. 1 Nested set

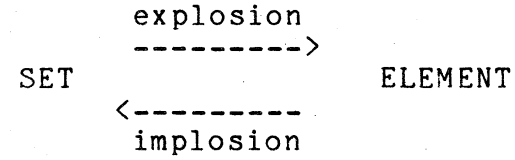
Basic operations to the nested set includes operations of conversion operations between set and element. By using these operations we can traverse from node to node in one to many mode. These are called 'implosion' operation and 'explosion' operation.

EXPLOSION(set)

explosion operation gives all belonging elements of a set.

IMPLOSION(element)

implosion operation gives all sets to which an element belongs.



With use of these operations along with normal set operations,

structure traverse between related nodes can be performed if we select one of the possible paths found by implosion or explosion.

3. Data Entry Language

In this section, we introduce a set of syntax of language for data entry. Databox do not expect any homogeneity in source data. Consequently, any type of information like record nor file can not be applied on them. Introducing language for describing indibidual data would work well except for efficiency in machine operations. Basic framework of the language is set to catch up common way of conveying information. It is selection of realities from possibilities. The possibilities aee conceived as class of information on which we can put label of a name as a single item. Actually, in its most simple form, information is represented as combination of the name of the class, i.e. type, and selected reality, instance. Without the information, we only suppose all of the possibilities. In this connection, the notion of attribut is regarded as a sort of the type in that certain entity is supposed to be exist as a 'test-body' for asking its aspects.

The type-instance coupling is recursively applied in this language. Each instance may be type-instance pair too. The syntax of the language is shown bellow.

<complex> ::= <type_name> = <instance>

<instance> ::= <atom> | <vector> | <tensor> | <set> | <complex>

<set> ::= ([<instance>]$_{n>0}$)

<tensor> ::= ([<vector> /]$_{n>1}$)

<vector> ::= "<" [<atom>]$_{n>1}$ ">"

<type_name> ::= charactor_string

<atom> ::= character_string | numerical_value

Next, we consider how the source data is translated into object data which is structured in nested set scheme.

Suppose a statement

hight = 170cm

is given. Where 'hight' is type_name and 170cm is instance. The statement can be expressed as a binary tuple if we assume that the frist element corresponds to type_name and second to instance. That is:

< hight, 170cm >

Exactly, the tuple should be

(<type_name, hight> , <instance, 170cm>)

in named set representation. As we have to avoid employing ordered set or named set, the tuples are converted into sets by introducing two special element '#TYPE' and '#INSTANCE' as system constants. Thus we get

((#TYPE, hight) , (#INSTANCE, 170cm))

This final form of the object data is illustrated in Fig. 2.

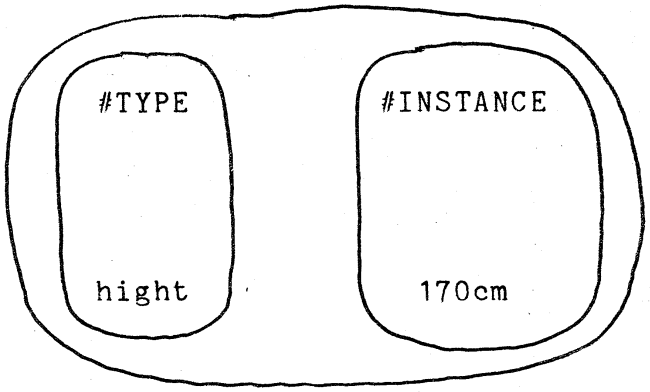


Fig. 2 Object data of 'hight = 170cm'

Let us see another example.

```
person = ( name = TARO,
           hight = 170cm,
           weight = 60kg,
           age = 30,
           programer,
           children = ( ( name = HANAKO, age = 3,
                          pets = (JOHN, TAMA ) ),
                        ( name = ICHIRO, age = 1 )
                      )
         )
```

where 'programer' is a simple code as an istance by which one can imagine what it means like in case of using keywords. Fig. 3 illustrates nested set stucture for the above example, where associations of items to constants, #TYPE and #INSTANCE, are

neglected.

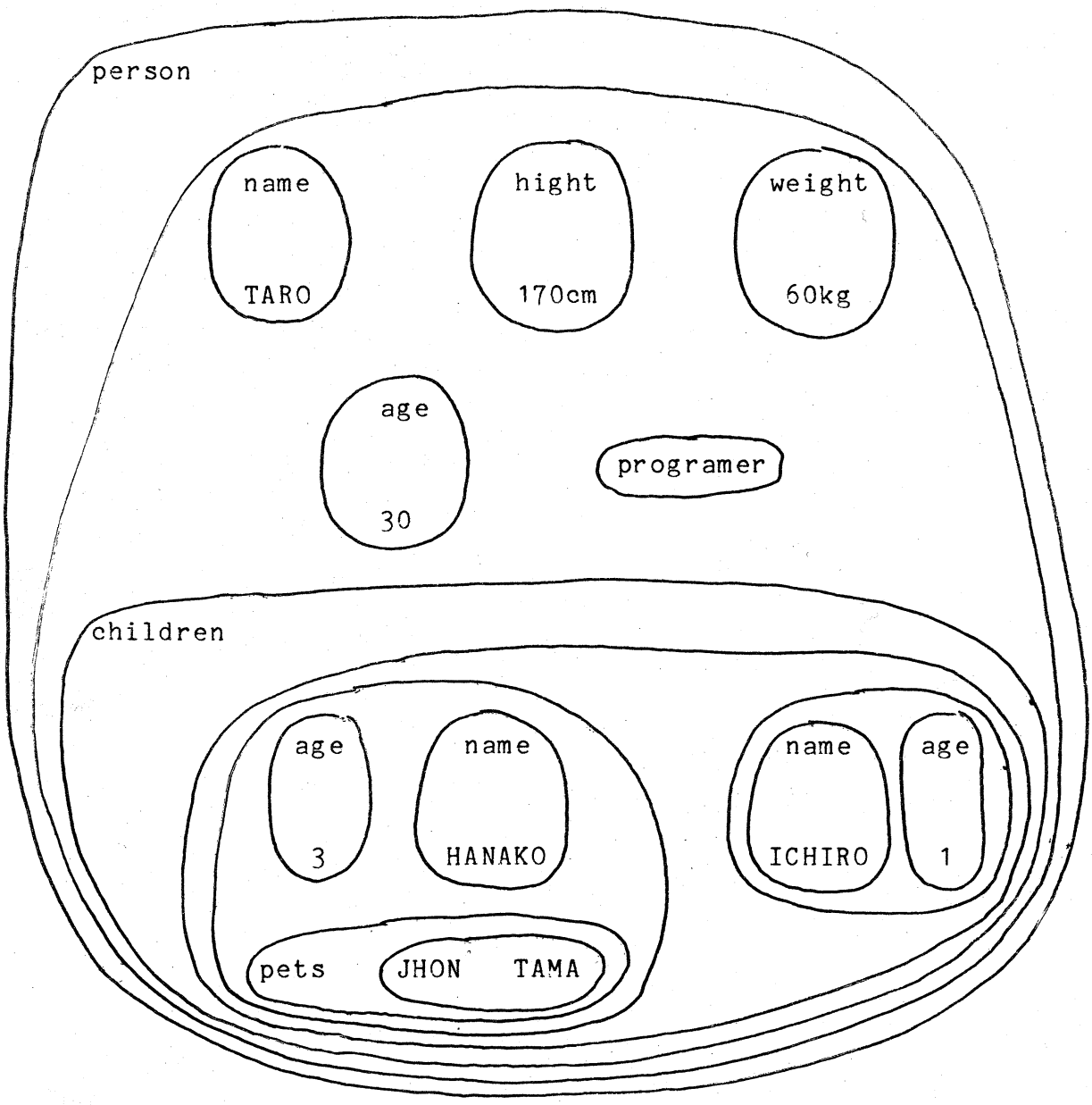


Fig. 3 An example of nested set

4. Data Manipulation

The object data is manipulated according to requirments of users. There are four basic types of manipulation; entry, update, deletion and query. Among the four, we focus our attention on the

query in this section. Before that, the first three are looked up.

(1) entry

Data entry operation is read as merge operation for a new nested set to master data. The procedure is:

(a) scan the lowest level nodes

(b) test matching of the nodes to primary set

if same node is found

then link to the node in the master.

else add and register the new node

(c) repeat (a) and (b) for next lowest nodes

(2) update

Delete-and-add sequence is applied for update operation on a node. However, it would likely to occur to modify a set by adding or eliminating the elements rather than to substitute whole the set including all of subordinate nodes.

(3) deletion

There are two alternatives for implementation of deletion operation. One is defered operation and the other is on the time operation:

(a) set obsolete flag on nodes to be deleted and reorganize whole sets after.

(b) delete all of subordinate nodes from a node which is specified to be deleted.

(4) query

Queries are expressed in form of type_instance pair in which keys for assosiation is included, i.e.

type_name = (type_name_instance_pair)

person = (hight = 170cm)

What the databox has to do is to find nodes containing the pair as one of its substructure. At first a set of nodes, "horizon", is filtered out by applying EXPLOSION operation on nodes of #TYPE and the first type_name of the query. Next, IMPLOSION and intersection operations are repeatedly applied on nodes for instances until the operations reach to the hirizon. Here we see how the procedure is actually performed for an example :

person = (hight = 170cm)

This query is trnasformed into object form as in Fig. 4

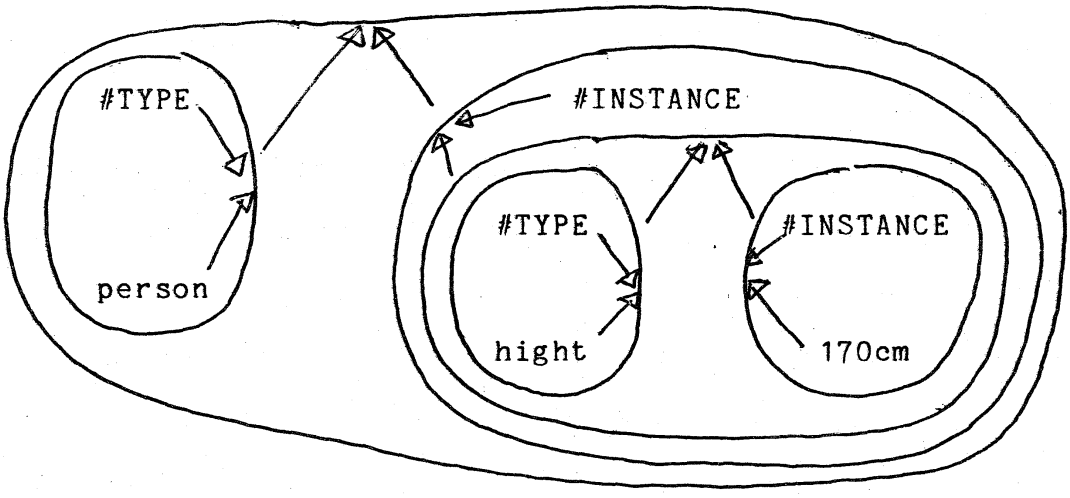


Fig. 4 Object form of a query 'person = (hight=170cm)'

Element to set expansion is performed by IMPLOSION to get the smallest common set of the nested set transrated from the query. For example, the next upper set of the set (#TYPE,

hight) is given by

impl(impl(#TYPE) & impl(hight)) .

In the same way, a set X containing the pair 'hight = 170cm' is defined as

X = impl(impl(#TYPE) & impl(hight)) &
impl(impl(#INSTANCE) & impl(170cm))

Finally, the answer Y for the query 'person = (hight=170cm)' is obtained by calculating

Y = impl(impl(#YTPE) & impl(person)) &
impl(impl(#INSTANCE) & impl(X))

5. Concluding Remarks

A new concept of information management system is proposed as databox. Databox is a personal information system which covers information in research activities. The distinctive point comparing with DBMS is that it doesn't assume any schema, instead that it employs a language for description of individual data. From point of view of data stucturing, the concept of access stucture is taken out of the system, like in case of relational model. It would be difficult to predict in what way the information is accessed especially within research environments. So that, actual path of access is derived from basic relationships among data.